

Création d'un outil de support informatique sur ligne.

Rapport de stage technicien Bachelor 2ème année chez
Ondal France - Mibelle Group.

Julien HUYGHE

Juillet 2021



1

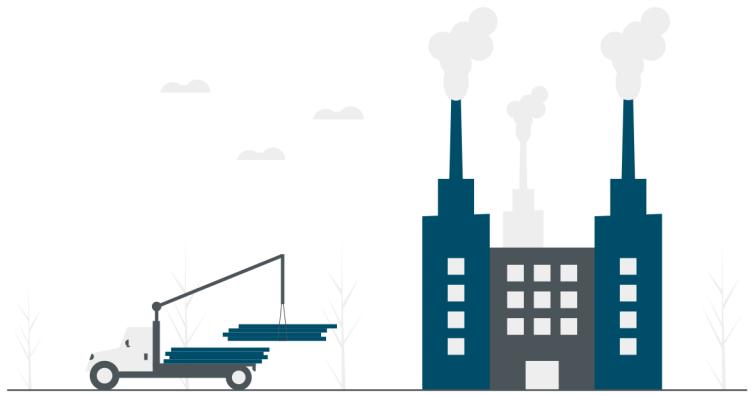


Table des matières

Remerciements	4
Introduction	5
Entreprise & Environnement	6
Identité de l'entreprise	6
Généralités	6
Historique	7
Création par le groupe Wella	7
Période creuse avec Procter & Gamble	7
Nouveau souffle du groupe Mibelle	7
Activités de l'entreprise	8
Mibelle Group	8
Entreprise Ondal	9
Organisation	10
Position dans l'organigramme	11
Réalisation de la mission de stage	12
Objectif de stage	13
Problématique	13
Périmètre du projet	14
Contraintes	16
Solutions envisagées	18
Solutions retenues	21
Résultats	22
Partie "app"	22
Partie "éditeur"	24
Problèmes rencontrés	26
Améliorations possibles	27
Conclusions	27
Annexes	27
Annexe 1 - Plan du site	29
Annexe 2 - Organigramme de l'entreprise	30

Liste des tableaux

1	Différents clients du groupe Mibelle.	9
2	Départements de l'entreprise Ondal.	10
3	Étapes de ma méthode de gestion PSDCA.	12
4	Fonctions de service et contraintes de notre système.	16
5	Bilan des avantages et inconvénients de la solution <i>Microsoft PowerApps</i>	18
6	Bilan des avantages et inconvénients de la solution <i>Python GUI</i>	19
7	Bilan des avantages et inconvénients de la solution <i>Microsoft Visual Studio</i>	20
8	Bilan des avantages et inconvénients à la création d'un site internet avec <i>ElectronJS</i>	21

Table des figures

1	Fiche d'identité de l'entreprise Ondal France	6
2	Illustration des valeurs du groupe Mibelle : la recherche de l'innovation, de la qualité, et son impact écologique.	8
3	Position dans l'organigramme	11
4	Schéma initial des étapes à suivre pour obtenir une procédure.	14
5	Diagramme FAST de la solution	15
6	Diagramme APTE de la solution	15
7	Schéma avec la solution des étapes à suivre pour obtenir une procédure.	17
8	Besoins implicites du système SPL	17
9	Illustrations de l'outil pour la partie "app" lors de scan d'étiquette.	22
10	Illustration de l'outil pour la partie "app" lors d'enregistrement de vidéo.	22
11	Illustration des technologies employées dans la partie "app".	23
12	Illustration de "SPL-editeur" pour générer des étiquettes.	24
13	Exemple d'étiquette pour le système SPL.	25
14	Plan du site	29
15	Organigramme de l'entreprise Ondal	30

Remerciements

Avant de commencer, je trouve normal de remercier plusieurs personnes qui ont contribué aussi bien à la réussite de ce stage, que mon développement professionnel et personnel, ainsi que l'écriture de ce rapport.

D'une façon générale, l'entreprise **Ondal France** appartenant à **Mibelle Group**, pour son accueil et dont les acteurs ont tous été bienveillant et pédagogue avec moi lorsque c'était nécessaire, et plus particulièrement;

- **Monsieur Laurent STEIN, mon maître de stage et responsable d'une ligne de production.** Au-delà de son encadrement et de ses conseils, il m'a fait confiance pour mener à bien cette mission. Sur un plan plus personnel, il m'a également fait découvrir la Moselle que je ne connaissais pas, département où se trouve l'entreprise.
- **Monsieur Pierre LAMY, alternant au service ressources humaines en charge des stagiaires.** Ses retours réguliers pour savoir si tout se passait bien, son sourire et son organisation ont contribué au fait que cette expérience a été très plaisante pour moi, et j'étais chaque jour enthousiasmé à l'idée de venir.
- **Nadya, Rachel, Bernadette, opératrices sur les lignes de production** dont les retours m'ont été précieux lors de cette mission.

De même, j'aimerais également remercier **Madame Carine Vignes, enseignante ENSAM et tutrice**, pour la re-lecture de ce rapport et son soutien lors de ma formation, sans qui ce stage n'aurait pas été possible.

Introduction



Je m'appelle **Julien Huyghe**, j'ai 22 ans, et je suis actuellement en deuxième année de **Bachelor de Technologie aux Arts et Métiers de Bordeaux-Talence**. J'apprécie cette formation pour la gamme de compétences qu'elle apporte et qui sont souvent mis en valeur lors de projets, permettant de concrétiser les compétences et connaissances acquises et forcant le travail en équipe.

De plus, j'ai deux passions ; l'une pour l'**aéronautique**, ce qui m'a conduit à passer ma licence de pilote privé ; l'autre pour l'**informatique**, pour les possibilités qui sont offertes et dont j'ai développé plusieurs solutions et outils au cours des dernières années.

Ainsi, dans mon idéal, il me faudrait trouver un stage en entreprise du milieu aéronautique, sur *Bordeaux*, en rapport avec l'informatique. Comme nous allons le voir au cours de ce rapport, on n'obtient pas toujours ce dont on souhaite, mais on est jamais à l'abris de (très) bonnes surprises.

Ce rapport rend compte d'une période de **stage en entreprise de douze semaines** effectuée au cours de cette deuxième année de formation **entre le 26 avril et le 16 juillet 2021**, dans une entreprise du nom de **Ondal France - Mibelle Group**, à Sarreguemines dans le magnifique département de la *Moselle*, en région *Grand-Est ex-Lorraine*.

J'ai obtenu ce stage sans aucune aide, relativement tôt avec cinq mois d'avance, au travers de la plateforme **JobTeaser**. La société **Ondal**, spécialisée dans la fabrication de cosmétiques et produits capillaires pour à la fois les professionnels et la grande distribution, recherche un stagiaire pour **développer un outil informatique pour des lignes de production**.

Au cours de ce rapport, nous aborderons ainsi **dans un premier temps l'entreprise et son environnement**; c'est-à-dire son histoire, ses valeurs et son organisation.

Par la suite, nous verrons plus en détails la **réalisation de la mission de stage**, avec la problématique initiale, les solutions et les résultats au terme de cette expérience professionnelle. Enfin, nous finirons par **plusieurs bilans vis-à-vis de ce stage** chez **Ondal** et ce que je peux en tirer.

Entreprise & Environnement

Identité de l'entreprise

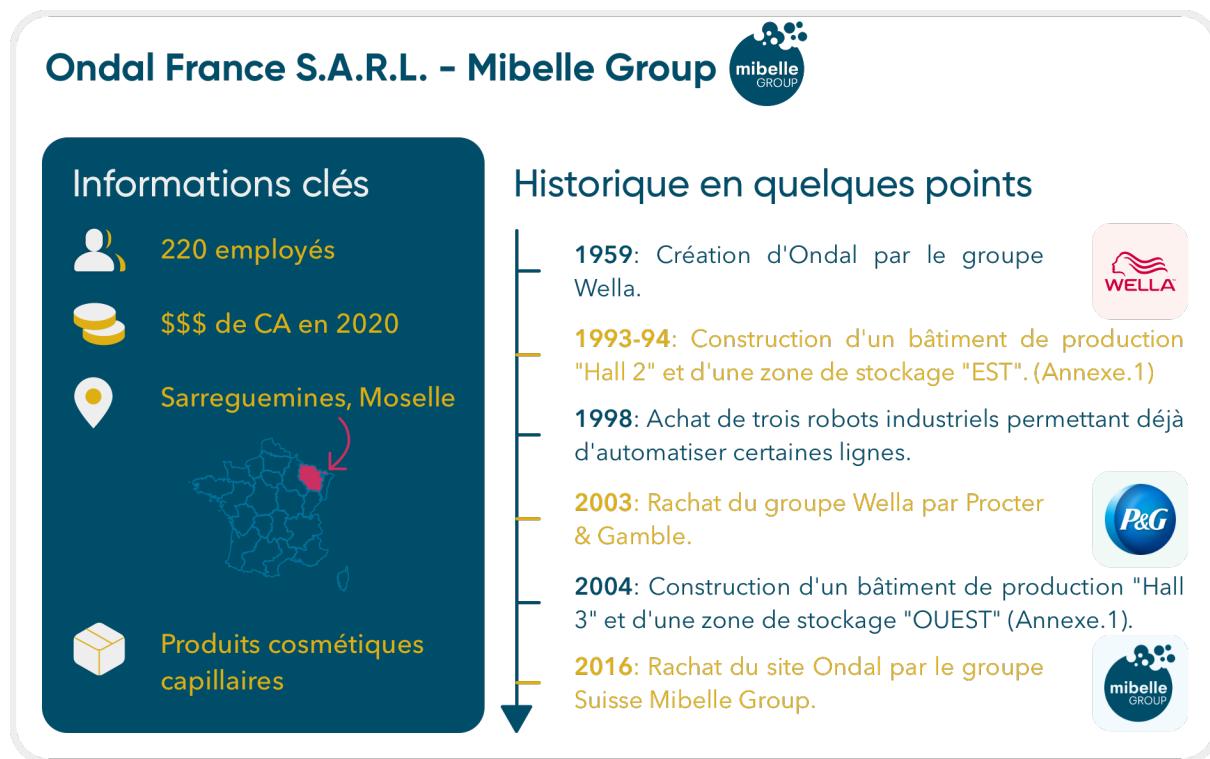


Fig. 1 : Fiche d'identité de l'entreprise Ondal France

Généralités

L'entreprise **Ondal France**¹ est une Société à Responsabilités Limitées (S.A.R.L.) située au 2 rue Denis Papin, à Sarreguemines, en Moselle région Grand-Est; et spécialisée dans la fabrication de produits cosmétiques pour le corps et les cheveux depuis sa création en 1959.

Au moment de l'écriture de ce rapport, l'entreprise est constituée de 220 employés répartis en plusieurs postes, dits de **matin** (6h-14h), **jour** (8h-16h), **après-midi** (14h-22h), et **nuit** (22h-6h), permettant d'avoir une continuité de fabrication. Parmi ces 220 employés "collaborateurs", environ 200 sont en CDI, le reste étant soit en CDD, intérimaire, ou alternant.

¹Dans un soucis de simplicité, je l'appellerai simplement **Ondal** par la suite.

Historique

Création par le groupe Wella

À l'origine, celle-ci a été créée par le groupe allemand Wella en 1959, et n'était constitué que d'un seul hall de production, le **hall 1**.² Les années 1990 ont montré une volonté forte de s'étendre ; avec le **hall 2** en 1992, le stockage «**est**» en 1993, et le rachat des locaux d'une entreprise voisine en 1996.

L'année 1998 marque l'acquisition de premiers robots industriels destinés à automatiser certaines tâches, comme de la mise en cartons de bouteilles. Enfin en 2000, la construction du magasin **ouest** est terminée.

Période creuse avec Procter & Gamble

À la suite du rachat du groupe **Wella** par **Procter & Gamble (P&G)** en 2003, l'entreprise subit plusieurs modifications aussi bien structurelles qu'organisationnelles. On passe d'une organisation pyramidale à une **organisation matricielle**³. L'usine se dote d'un nouveau hall de production d'environ 4000m², le **hall 3**, qui est à ce jour le dernier, et le plus gros. Enfin, une première politique de développement durable est mise en place, avec un traitement local de toutes les eaux usées ainsi qu'un tri très sélectif des déchets.

Nouveau souffle du groupe Mibelle

Entre le milieu des années 2000 et 2016, l'entreprise est dans une situation plutôt mauvaise. Tensions sociales, manque d'investissements et problèmes d'organisation sont présents. De gros changements sont arrivés en 2016 suite au rachat de l'entité Ondal et de ses locaux par le groupe **Mibelle Group**⁴, entreprise du groupe Suisse **Migros**. Encore une fois, l'organisation interne est modifiée, notamment avec certains services et outils. De plus, une politique de développement durable et du bien-être humain est adoptée, considérant à présent que les employés sont le cœur de l'entreprise. Ces changements ont contribué à une croissance de 10% par an depuis 2016.

²Voir le plan de l'usine situé en annexe 1.

³Voir l'organigramme des employés en annexe 2.

⁴Encore une fois, j'utiliserai soit le terme **Mibelle**, soit **groupe Mibelle** lors de ce rapport.

Activités de l'entreprise

L'entreprise Ondal appartient à l'entreprise Mibelle Group, elle-même entreprise du groupe Migros. Ondal France constitue uniquement l'entité juridique ; elle ne dispose même pas de logo. Ainsi, il est plus intéressant et important de présenter l'entreprise Mibelle et non l'entité Ondal France.

Mibelle Group

Mibelle est une entreprise appartenant au géant de la distribution Suisse **Migros**, fabricant de produits cosmétiques et d'hygiène aussi bien pour la marque Migros, que pour des marques tierces. Il existe également une filiale, Mibelle *Biochemistry*, dédiée à la recherche et au développement d'ingrédients actifs pour des applications cosmétiques.

À ce jour, le groupe Mibelle comporte 1 250 employés de 36 nationalités différentes, avec une parité de 46% de femmes et 54% d'hommes. Répartie sur six sites en Suisse, France, Grande-Bretagne et Corée du Sud, elle a réalisé un chiffre d'affaires de 650M de francs suisse (CHF) en 2019.

Telle qu'elle se décrit, ses valeurs sont autour de la **passion** à performer et s'améliorer, son **engagement** envers l'environnement et le respect de ses clients, sa **volonté** d'innover en continu, et son **impact** pour contribuer au bien-être de ses clients sur la durée.



FIG. 2 : Illustration des valeurs du groupe Mibelle : la recherche de l'innovation, de la qualité, et son impact écologique.

Mibelle est décomposé en quatre branches;

- **Personal Care & Beauty** : Produits cosmétiques
- **Home Care** : Produits d'hygiène et d'entretien
- **Nutrition** : Produits et graisses alimentaires
- **Biochemistry** : Principes actifs

L'entreprise Ondal, se trouve dans le branche *Personal Care & Beauty*, qui conçoit, fabrique et conditionne des produits cosmétiques se voulant respectueux de l'environnement et du consommateur. Ainsi, ses clients sont aussi bien professionnels que personnels;

TAB. 1 : Différents clients du groupe Mibelle.

Client	Description
Professionnels	Produits adaptés à des besoins spécifiques pour leur activité comme les soins bucco-dentaires.
"MBO" (Multiple Brand Owners)	Groupes possédant plusieurs marques et dont des professionnels peuvent être clients, comme des coiffeurs.
Détaillants	Marques propres vendant au détail.
Consommateurs	Produits semi-professionnels destinés à la grande consommation.

Entreprise Ondal

Comme présenté précédemment, l'entreprise Ondal est spécialisée dans la fabrication de produits cosmétiques pour le corps et les cheveux depuis sa création.

Sur site, arrive les matières premières des différents produits fabriqués, le principal étant le **péroxyde d'hydrogène**. Plusieurs services se relaient, pour créer les **jus**⁵, les conditionner, et les emballer prêt pour l'expédition.

⁵Le(s) **jus** est le terme utilisé pour désigner le produit contenant les principes actifs.

Organisation

L'entreprise Ondal est organisée selon un modèle {modèle organisation}. Comme dans toute entreprise structurée, il existe des services (ou départements), qui sont chacun en charge d'une partie précise du fonctionnement de l'entreprise. Ces différents départements sont répertoriés dans le tableau ci-dessous ;

TAB. 2 : Départements de l'entreprise Ondal.

Département	Fonction et description
Engineering	Conception, développement et protection des technologies, systèmes, et bâtiments de production.
Finances	Prédiction et gestion des coûts de production et de fonctionnement de l'entreprise.
Logistique/Planning	Suivi et gestion des flux pour fournir les différents services et clients.
Making	Fabrication des jus à partir des matières premières.
MPD ⁶	Standardisation et optimisation des formules, emballages, processus, matériaux, équipements et procédures.
Packing	Conditionnement des produits.
Qualité	Vérification et recherche d'amélioration concernant la qualité des formules et processus de fabrication.
Ressources Humaines	Gestion et coordination des différents employés de l'entreprise.

Ces différents départements sont hiérarchisés de telle sorte que chaque branche soit indépendante en terme de gestion. Les gestionnaires, sont alors appelés des "**LMT**", pour *Leader Management Team*.

Ces LMT, rendent ensuite compte au PDG de l'entreprise ; Monsieur *Gaëtan PERZO*.

⁶MPD signifie **Materials, Process & Delivery**.

Position dans l'organigramme

Lors de la section suivante concernant l'activité en entreprise, on verra que je suis amené à travailler avec de la documentation existant majoritairement dans le département "**Packing**". Bien que le besoin et la solution peuvent être étendus à tous les départements, j'ai majoritairement travaillé avec les personnes du département Packing ; mon tuteur de stage est un des responsables d'une ligne d'assemblage.

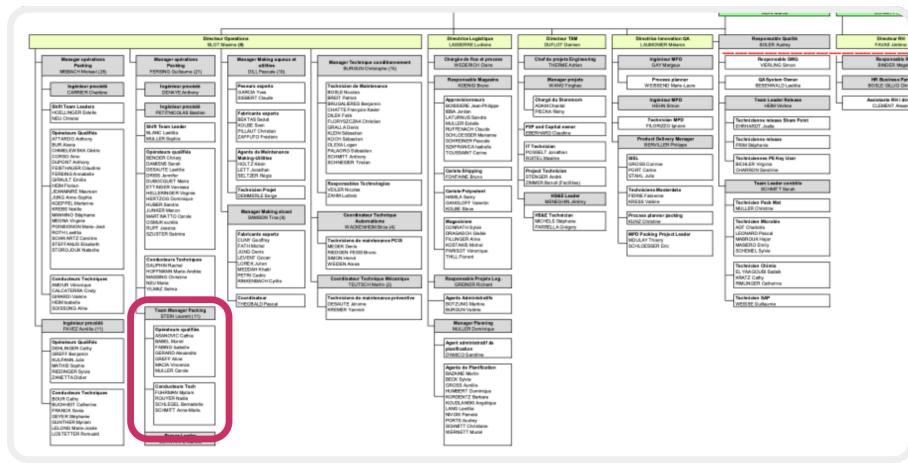


FIG. 3 : Position dans l'organigramme

Réalisation de la mission de stage

Au cours de cette section, nous allons aborder le travail effectué au cours du stage en lui-même. Nous allons partir d'une présentation de l'existant, de la problématique lors de mon arrivée en entreprise et du besoin qui en découle; pour ensuite voir comment une solution a pu être imaginée et si elle permet de répondre à ce besoin.

Dès mon arrivée, Monsieur Laurent STEIN, mon tuteur de stage, m'a guidé pour utiliser une méthode de gestion de projet : le **PDCA**. Je l'ai légèrement adapté pour obtenir la méthode **PSDCA**, à cause du temps limité lors de ce stage de 12 semaines. À la place d'essayer de potentielles solutions, je choisis directement une solution à partir du périmètre du projet, qui me servira de fil rouge tout le long du projet.

TAB. 3 : Étapes de ma méthode de gestion PSDCA.

Étape	Description
Plan	Identification et établissement du périmètre du projet.
Seek	Recherche de solutions et bilan faisable / non-faisable.
Do	Développement de la solution retenue.
Check	Mise en production et recherche de résultats.
Act	Amélioration et optimisation à partir des résultats.

La première étape étant la planification du projet, il convient de définir dès maintenant plusieurs éléments ; l'objectif de stage, et la problématique. Ces deux éléments nous permettront de définir le périmètre du projet, de déterminer les contraintes pertinentes et de trouver une solution proposant le meilleur compromis.

Objectif de stage

Tel qu'il a été défini dans la convention de stage, l'objectif (ou l'intitulé) est le suivant :

Création d'un support informatique sur les procédures de travail sur ligne.

Bien que cet intitulé soit assez explicite, il peut être ambiguë ; Un support physique ou logiciel ? Qu'est-ce qui est défini comme une procédure de travail sur une ligne ?

Pour simplifier les choses, il est question lors de ce rapport de **créer un logiciel-outil informatique permettant d'accéder à des documents de maintenance et de formation de divers équipements présents sur des lignes de production.**

Problématique

Comme vu lors de la présentation de l'entreprise, Ondal est un site de production de produits. Afin d'assurer le bon fonctionnement de l'usine et une bonne qualité de produits, il est nécessaire d'instaurer des procédures uniformisant les actions à mener. Le but est que n'importe qui, après avoir lu une procédure, soit capable d'effectuer les actions qui y sont décrites. Par exemple, où et comment lubrifier un rail de guidage d'une remplisseuse de bouteilles.

Ces procédures, servent alors à la fois de formation pour les nouveaux personnels, et de rappel si des opérations plus complexes sont nécessaires.

Dans l'état actuel des choses, ces procédures sont des feuilles de données *Excel* stockées dans des dossiers sur un disque réseau partagé. Afin de les consulter, il est fortement recommandé d'utiliser des bases de données *Access* créées pour. Celle contenant les procédures de formation sont dans la base de données **eQualif**, les autres dans la base **eJOBOP**.

Le soucis, c'est qu'il est très souvent fastidieux et compliqué d'accéder à une procédure en particulier. Par exemple, la procédure associée au vidage de la mémoire tampon d'une trieuse pondérale de la ligne **SG41**, a comme entrée de titre "*Reset Checkweigher*", tandis que toutes les autres trieuses pondérales ont un nom différent.

De plus, cette base de données est uniquement accessible sur un ordinateur souvent éloigné de l'équipement. Ainsi, cela représente une perte de temps pour à la fois se déplacer sur un ordinateur, mais aussi pour rechercher une procédure en particulier.

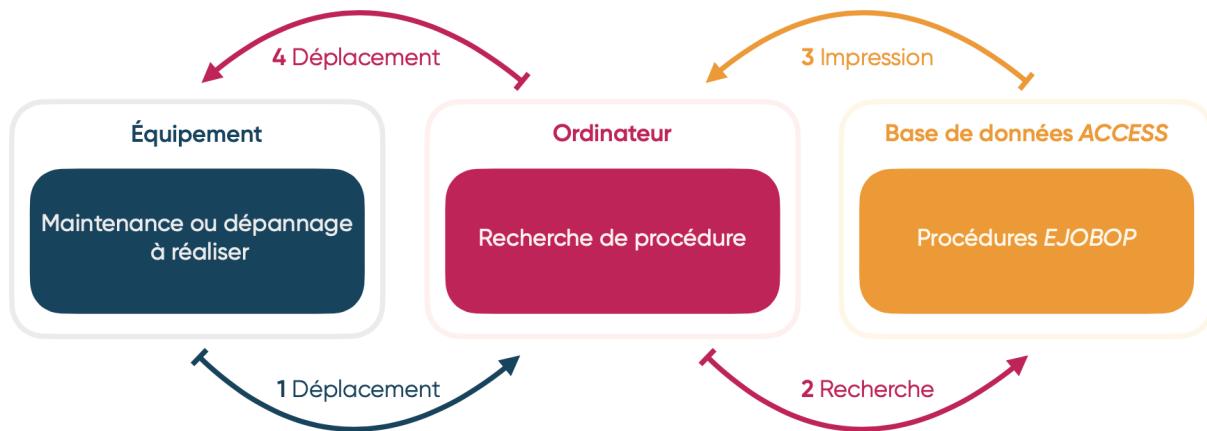


FIG. 4 : Schéma initial des étapes à suivre pour obtenir une procédure.

Le besoin est alors plutôt clair : il s'agit de **trouver une solution pour automatiser et améliorer la recherche et l'accès des procédures pour un équipement donné**.

Périmètre du projet

Une fois le besoin identifié, il nous reste à établir le périmètre du projet pour avoir une idée claire de la solution à envisager.

L'outil est en majorité destiné aux opérateurs et techniciens sur une ligne de production. On peut dès lors déterminer les acteurs comme suit :

- **Cibles** : Toute personne travaillant sur ligne de façon régulière ou non.
- **Fonction** : Améliorer et automatiser l'accès à des procédures d'utilisation, maintenance, formation et réparation disponibles sur un disque en réseau partagé.
- **Matière d'œuvre** : Ligne de production sur site.
- **Produit & service** : Service de Procédures sur Ligne (**SPL**).

À partir de ces éléments, il est possible d'utiliser les outils APTES et FAST pour visualiser les contraintes nécessaires en vue de l'établissement du cahier des charges ;

Diagramme FAST L'outil FAST nous permet d'associer les différents éléments de notre périmètre de projet, pour en obtenir une vue d'ensemble en vue de la recherche de contraintes auxquelles doit répondre notre solution.



FIG. 5 : Diagramme FAST de la solution

Diagramme APTE Les points clés pris en compte, sont l'**ergonomie** en vue de l'utilisation de l'outil par un travailleur sur ligne, la **connectivité** et l'**intégration avec son environnement**, ainsi que l'**automatisation** du processus.

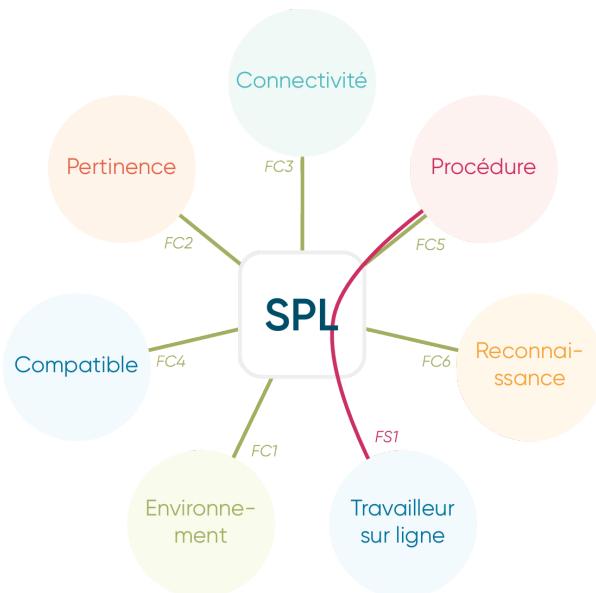


FIG. 6 : Diagramme APTE de la solution

TAB. 4 : Fonctions de service et contraintes de notre système.

Fonction	Description	Flexibilité
FS1	Permettre l'automatisation et améliorer l'accessibilité des procédures sur une ligne de production.	S0
FC1	Doit pouvoir être utilisé en environnement industriel.	S1
FC2	Doit fournir une procédure pertinente à la demande.	S0
FC3	Doit pouvoir récupérer uniquement les procédures disponibles sur un disque réseau partagé.	S0
FC4	Doit être compatible avec n'importe quelle ligne de production.	S1
FC5	Doit être compréhensible et utilisable par un travailleur sur la ligne.	S1
FC6	Doit pouvoir identifier la demande rapidement et automatiquement.	S1

Contraintes

Nous pouvons maintenant nous intéresser aux contraintes à la fois généralistes et techniques de notre projet;

- Système **portable**, avec recours à l'utilisation de tablettes numériques sous Windows.
- **Ergonomique**, qui nécessite pas ou peu de formation et soit simple à l'utilisation.
- **Compatible avec les infrastructures existantes** comme les disques en réseau.
- "**Future-proof**", utilisant des technologies dites "LTS"⁷, n'ayant pas de dépendances externes et des versions bloquées.
- **Technologies en libre-accès** ou sous licence dont dispose l'entreprise, comme le pack Office 365.
- **Processus automatisé**, avec utilisation de codes-barres disposant de la procédure encodée.

Licences Un détail auquel je n'avais pas initialement pensé concerne les licences utilisées. En tant que particulier, je ne suis pas amené à réellement m'en occuper puisque la totalité des logiciels et outils que j'utilise n'ont pas pour vocation à être re-distribués à des personnes tierces. Cependant, dans le cadre professionnel, il est nécessaire de prendre en compte les licences utilisées pour développer un outil.

⁷LTS signifie **Long Term Support**.

Automatisation Comme exposé précédemment, une des contraintes de notre solution est l'automatisation. Celle-ci m'a été guidé par mon maître de stage, qui m'a imposé l'utilisation d'étiquettes avec un code-barre disposant du nom de la procédure à rechercher sur le réseau. Ces étiquettes seront placées directement sur l'équipement concerné.

L'objectif est d'obtenir une solution qui corresponde au diagramme suivant.

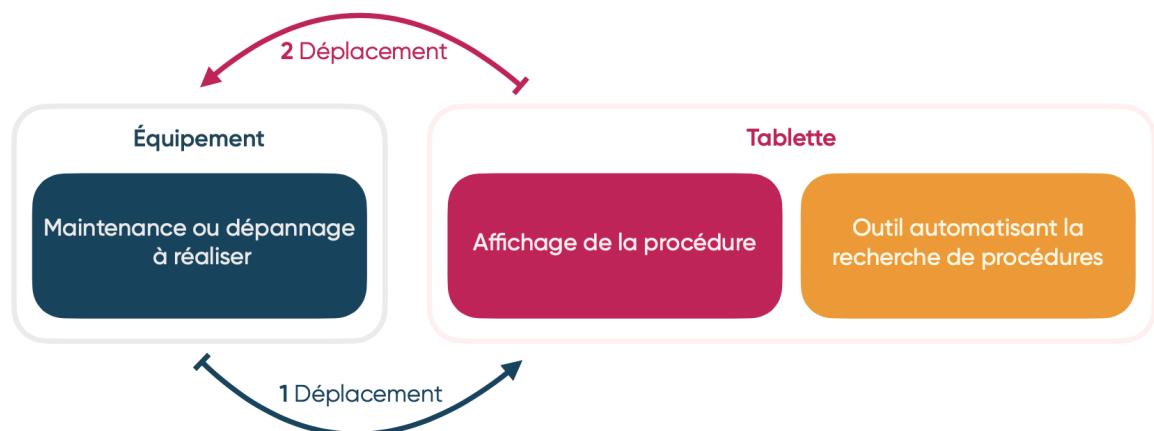


Fig. 7 : Schéma avec la solution des étapes à suivre pour obtenir une procédure.

Pour avoir une fonctionnalité d'automatisation, l'idée retenue est la création d'étiquettes disposant d'un QR Code (donc un code-barre en deux dimensions) permettant de guider vers une procédure en particulier. Ainsi, on peut déterminer deux besoins implicites à notre système "**SPL**", les deux ayant le même cahier des charges ;



Fig. 8 : Besoins implicites du système SPL

Solutions envisagées

Maintenant que nous avons les besoins, le périmètre de notre projet et nos contraintes, nous allons nous intéresser à la recherche de potentielles solutions permettant de répondre à nos deux besoins. Les solutions envisagées ont été trouvées à partir de connaissances personnelles et de recherche sur internet.

Microsoft PowerApps



La première solution envisagée est l'utilisation de l'application **Microsoft PowerApps** contenu dans le pack *Office 365*.

Je ne connaissais pas cet outil. Pour expliquer simplement, il permet de **créer des applications avec une interface utilisateur sans programmation pour traiter des données**.

Très facile d'utilisation et d'apprentissage, il y a quelques points rédhibitoires. Le premier étant lié à notre contrainte de développer pour une tablette fonctionnant sous Windows. Bien qu'il existe un module permettant de scanner des codes-barres, il n'est disponible de façon stable que sous *Android* et *iOS*. De plus, il est impossible de demander à ouvrir des fichiers sur un disque local partagé sans bidouiller une solution via des protocoles *URI*, toutes les données se devant d'être en ligne via *SharePoint* ou *OneDrive*. Pour ces deux raisons, on ne peut retenir cette solution.

TAB. 5 : Bilan des avantages et inconvénients de la solution *Microsoft PowerApps*.

Avantages	Inconvénients
Création d'interface sans programmation.	Données en ligne "cloud" uniquement.
Intégré dans le pack <i>Office 365</i> dont dispose l'entreprise.	Fonctionnalité d'interprétation de codes-barres uniquement sous <i>Android</i> et <i>iOS</i> .
Lecture native et facile de codes-barres.	-

Python GUI



Une deuxième solution est de développer une application de bureau sous Python en utilisant soit le module *tKinter*, soit *PyQt5* pour se charger de l'interface utilisateur.⁸ L'avantage est d'avoir un **programme très léger et relativement simple de fonctionnement**. De plus, il aurait été possible d'utiliser des outils comme *cx_Freeze*, pour empaqueter le fichier Python avec toutes ses dépendances et obtenir un fichier exécutable directement avec n'importe quel ordinateur Windows.

Cependant, la structure de la solution et le fait de devoir bricoler des fonctionnalités pour faire ce dont elles ne sont pas conçues pour à la base, m'a fait remettre en question la validité de cette solution, et j'ai choisi de ne pas la retenir au final pour la partie "**app**".

En revanche, les avantages cités sont parfaits dans l'optique de générer des étiquettes de procédure. Ainsi, c'est la solution retenue pour la partie "**éditeur**" du système **SPL**.

TAB. 6 : Bilan des avantages et inconvénients de la solution *Python GUI*.

Avantages	Inconvénients
Programmation simple.	Aucune fonctionnalité réellement adapté au besoin, comme par exemple l'affichage d'un flux vidéo et le décodage continu de l'image. pour rechercher un code-barre.
Open-source via la <i>Python Software Foundation (PSF)</i>	Structure mono-fichier nuisant à la lisibilité du projet.
Programme final très léger	-

⁸On va définir deux termes : **UI** pour *User Interface*, et **GUI** pour *Graphical User Interface*. Ces deux termes représentent la partie graphique de l'interface utilisateur.

Microsoft Visual Studio



Une troisième solution concerne l'utilisation de Microsoft Visual Studio, logiciel de Microsoft permettant le développement de logiciels en langages C++ et C#. **Des fonctionnalités légères et rapides dites "natives", permettant de faire ce que l'on veuille avec un minimum de contraintes.** Cependant, celui-ci appartenant à Microsoft, il est propriétaire et ni l'entreprise Ondal, ni Mibelle Group, ne disposent de la licence. Ainsi, il n'est pas possible de l'utiliser. De plus, celui-ci est plutôt lourd et long d'apprentissage pour pouvoir modifier le programme final. Pour ces raisons, je ne l'ai pas retenu.

TAB. 7 : Bilan des avantages et inconvénients de la solution *Microsoft Visual Studio*.

Avantages	Inconvénients
Très complet et permet d'obtenir la solution idéale.	Licence propriétaire.
Programme final très léger et performant.	Long d'apprentissage.

Site internet et ElectronJS

La dernière solution envisagée, est le développement d'un site internet classique en *HTML, CSS et Javascript*, qui serait empaqueté avec un outil du nom d'*ElectronJS*. Simplement, *ElectronJS*, basé sur une solution *Chromium* avec un moteur de rendu *Blink*, permet d'afficher un site internet comme un logiciel classique, mais en apportant des fonctionnalités normalement pas disponibles à un site internet, comme l'accès au système de fichiers, le lancement d'applications, etc. Des exemples d'applications utilisant *ElectronJS*, sont par exemple *Microsoft Teams*, *Slack*, *Skype* ou *Microsoft Visual Studio Code* (utilisé pour l'écriture de ce rapport au passage).



En termes de fonctionnalités, on peut encore faire ce que l'on souhaite avec sans avoir à se soucier de contraintes particulières. De plus, on utilise des langages simples d'apprentissage et d'utilisation, que sont HTML, CSS et Javascript. Le point noir de cette solution, est la taille ainsi que les performances du programme final. Sur un ordinateur, cela ne pose pas vraiment de soucis, mais sur une tablette peu performante, l'utilisation peut être plutôt lente.

TAB. 8 : Bilan des avantages et inconvénients à la création d'un site internet avec *ElectronJS*.

Avantages	Inconvénients
Énormément de fonctionnalités avec beaucoup de documentation.	Programme final plutôt lourd.
Langages simples d'apprentissage et d'utilisation.	-
Modules tout-fait, comme la détection de code-barre.	-

Solutions retenues

À partir des solutions exposées, la solution retenue pour la partie "**app**" est le développement d'un site internet avec l'utilisation d'*ElectronJS*. Par rapport aux autres solutions, le programme terminé sera plutôt lourd, mais cela n'est pas une contrainte de projet. En revanche, il satisfait les autres contraintes, étant simple d'utilisation et facilement reprenable, libre et open-source, et permettant l'automatisation de certaines tâches.

La partie "**éditeur**" sera basée sur une solution *Python GUI*, du fait de la simplicité et du besoin de générer un QRCode. Il n'y a pas de nécessité de créer un outil permettant à la fois de générer et d'interpréter des étiquettes ; cela alourdirait et ralentirait le programme final.

Au cours de la prochaine section, nous aborderons le fonctionnement de ces deux solutions.

Résultats

Partie "app"

Pour la partie "app", le choix a été fait de développer notre outil comme pour un site internet, puis d'utiliser *ElectronJS* pour le rendre utilisable comme un logiciel classique.

La principale fonction de ce logiciel, est de permettre de scanner des étiquettes avec un code QR faisant lien vers une procédure en particulier. Une interface graphique a été ajoutée à celui-ci afin de rendre l'utilisation simple et ergonomique, comme présenté ci-dessous ;



Accueil

Recherche de code QR

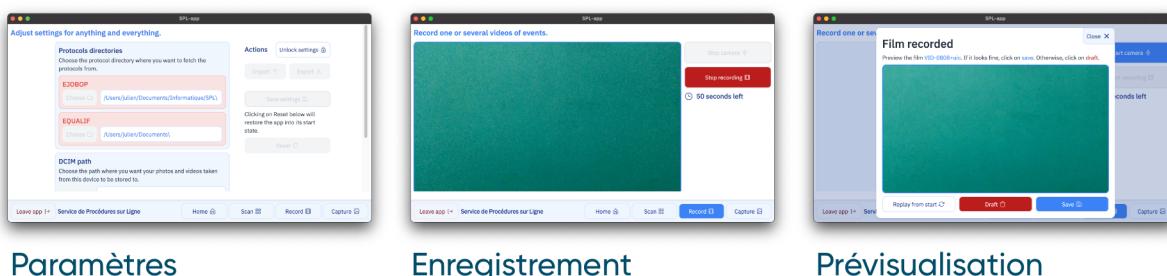
Résultat du code QR

FIG. 9 : Illustrations de l'outil pour la partie "app" lors de scan d'étiquette.

L'étiquette à été volontairement cachée dans le but d'illustrer le phénomène de scan d'étiquette.

Les captures d'écran ont été prises sous macOS, mais le fonctionnement sous Windows est strictement le même.

De plus, ce support permet la mise en place de fonctionnalités supplémentaires ; imaginons que quelqu'un travaillant sur la ligne remarque un problème. Cette personne enregistre une vidéo, qui est sauvegardée sur le disque en réseau partagé. Le manager de la ligne a donc un support visuel aidant à résoudre le problème.



Paramètres

Enregistrement

Prévisualisation

FIG. 10 : Illustration de l'outil pour la partie "app" lors d'enregistrement de vidéo.

Technologies et fonctionnement La partie “app” se base sur des technologies exclusivement open-source disponibles sur *GitHub*.



SvelteJS



Tailwindcss



ElectronJS



Node.js

Fig. 11 : Illustration des technologies employées dans la partie “app”.

- **SvelteJS** est un *framework*⁹ Javascript destiné à convertir une partie structurelle (HTML), interactive (Javascript) et stylistique (CSS), en un seul fichier Javascript. On dispose d’outils permettant de faciliter le développement, ainsi que d’un fichier Javascript optimisé en sortie.
- **TailwindCSS** est un autre framework Javascript, cette fois-ci pour le style au format CSS. Il a la même fonction que SvelteJS, mais pour obtenir un fichier de style CSS.
- **ElectronJS** permet d’empaqueter notre site internet, et d’y intégrer Node.js pour pouvoir être exécuté comme un logiciel classique.
- **Node.js** fournit des outils pour accéder à fonctions dont on n’aurait pas accès normalement, comme l’écriture de fichier sur disque, ou l’exécution de commandes.

Pour finir, on peut récupérer le flux vidéo d’une webcam en utilisant un *API*¹⁰ qui se nomme *Navigator*, et plus particulièrement *Navigator.mediaDevices*. En quelques lignes, on réussi à obtenir un flux vidéo avec des paramètres souhaités.

⁹Un framework peut être vu comme un ensemble d’outils qui permettent de définir une structure à un logiciel, tout en facilitant le développement de celui-ci.

¹⁰**API** est un terme pour *Application Programming Interface*, représentant un ensemble normalisé de classes, de méthodes, de fonctions et de constantes qui sert de façade par laquelle un logiciel offre des services à d’autres logiciels.

Partie "éditeur"

La partie "**éditeur**" est développée à l'aide de Python et de son module **GUI**¹¹ intégré *tKinter*. Le fichier source ainsi que ses dépendances, sont empaquetés pour être executé depuis n'importe quel ordinateur à l'aide de *cxFreeze*. De plus, le module *qrCode* et *docxtpl* sont utilisés, respectivement pour générer un code QR, et générer un fichier au format Word *.docx* à partir d'un fichier source dit "template".

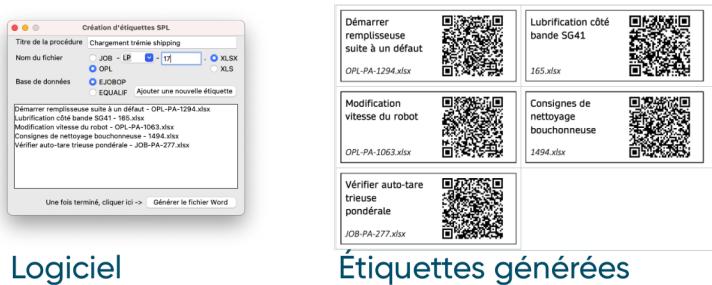


FIG. 12 : Illustration de "SPL-editeur" pour générer des étiquettes.

Concernant l'interface en elle-même, il y a plusieurs contraintes à prendre en compte ;

1. Nous avons deux bases de procédures distinctes ; *EJOBOP* et *EQUALIF*.
2. Les procédures sont soit des "aides au travail" (JOB), soit des "consignes en un point" (OPL).
3. Chaque département dispose d'un code à deux lettres. Exemple pour "packing" : PA.
4. Les procédures disposent d'un numéro unique au sein d'un même département.
5. Les procédures sont en majorité au format ".xlsx", mais il en existe encore au format ".xls".

Pour une procédure de la base *EJOBOP*, un exemple de nom de procédure est donc : **JOB-PA-1234.xlsx**.

Pour une procédure de la base *EQUALIF*, seul un identifiant unique est utilisé en plus de l'extension. Par exemple : **1234.xlsx**.

¹¹On va définir deux termes : **UI** pour *User Interface*, et **GUI** pour *Graphical User Interface*. Ces deux termes représentent la partie graphique de l'interface utilisateur.

Étiquette À cause des contraintes citées précédemment, il est nécessaire d'avoir un format d'étiquettes particulier.



Fig. 13 : Exemple d'étiquette pour le système SPL.

Sur cette page, l'étiquette est au format 1 : 1.

Sur l'étiquette, nous pouvons retrouver trois éléments :

- Un titre permettant la compréhension par un être humain.
- Le nom de la procédure encodée dans le code QR pour disposer d'une redondance en cas de soucis.
- Un code QR avec le nom de la procédure encodée.

Le code QR est de type Model 2, avec un nombre de modules automatique, disposant d'une correction d'erreur jusqu'à 30% ; pratique dans un environnement plutôt sale comme à proximité d'une machine.

Puisque nous pouvons afficher des procédures depuis à la fois la base *EJOBOP* et *EQUALIF*, il est également nécessaire d'encoder le nom de la base en plus du nom de la procédure. Le plus simple pour y parvenir, est l'utilisation d'un format **clé : valeur**, comme le format JSON¹². Par exemple, le code QR de l'étiquette en Figure 13 encode ces informations ;

```
{  
    "BASE": "EJOBOP",  
    "PROCEDURE": "JOB-PA-277.xlsx"  
}
```

De plus, ce format est bien pratique puisqu'il permet la conversion direct de son contenu en objet Python ou Javascript.

¹²Le format JSON (JavaScript Object Notation), est un format texte permettant de structurer des données en utilisant le modèle clé : valeur.

Problèmes rencontrés

D'une façon générale, le développement de ces deux solutions a été fluide et sans réel soucis, tant en termes de compétences de programmation, que de soucis techniques. Cependant, certains points sont intéressants à relever ;

- **Partie "app"** :

1. Le gestionnaire de modules *NPM*¹³ de *Node.js* présentait des **soucis de certificats pour télécharger les paquets demandés**. Cela était dû au proxy de l'entreprise qui bloquait certains fichiers et adresses. Une solution a été de désactiver la vérification d'authenticité du certificat du serveur de téléchargement, en utilisant le protocole HTTP plutôt que HTTPS. Une autre solution, et celle qui a été apportée car l'autre présente des soucis de sécurité, a été de **changer pour une connexion internet ayant un proxy autorisant ces actions**.
2. Comme évoqué précédemment, les licences utilisées sont importantes. Ainsi, il a fallu **lire et vérifier les droits de chaque logiciel**, et trouver des alternatives si besoin.
3. Sur un point plus technique, **l'API¹⁴ d'ElectronJS et de Node.js n'est disponible que pour le script de pré-chargement de l'application**, pour des questions de sécurité. Ainsi, il n'était pas possible d'utiliser les fonctionnalités de *Node.js* sur les pages de l'application, ce qui ne permettait pas le développement de celle-ci. La solution a été de **spécifier dans les paramètres d'ElectronJS d'autoriser l'utilisation de son API depuis n'importe où**. Ça n'a pas été compliqué, mais j'y ai passé du temps. (c.f. *contextIsolation*)

- **Partie "éditeur"** :

1. La grosse difficulté concernant la partie "éditeur", a été la **structure des données**. Le module *docxtpl* permettant de générer un document Word à partir d'un fichier template, accepte un objet unique. De plus, celui-ci comporte deux boucles itératives POUR. On se retrouve donc avec **un dictionnaire, ayant deux listes, ayant un objet**. Malgré tout, cela reste le moyen le plus simple de structurer les données.
2. Une autre difficulté, a été la **création du GUI**. Je ne connaissais pas du tout le module *tKinter* avant, il a donc fallu m'adapter. Le gros soucis a été le **positionnement des différents éléments en coordonnées absolues**. Encore une fois, rien de compliqué, mais cela a pris un peu de temps.

¹³**NPM** pour *Node Package Manager*, est un gestionnaire de modules (ou paquets) intégré à *Node.js*.

¹⁴**API** est un terme pour *Application Programming Interface*, représentant un ensemble normalisé de classes, de méthodes, de fonctions et de constantes qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.

Améliorations possibles

On verra dans la partie suivante que la solution remplit pleinement le cahier des charges. Cependant, j'estime que l'on peut toujours améliorer un produit. Ainsi, avec plus de temps et de moyens, il serait intéressant ;

- **Traduire l'interface.** Beaucoup de gens parlent allemand, ou du moins le patois lorrain : le platt. Traduire l'interface serait un gain en plus d'ergonomie.
- **Introduction de vidéos pas-à-pas.** Le but de la solution est d'aider les utilisateurs avec les procédures. À l'instar d'un tutoriel vidéo, on pourrait introduire des vidéos pas-à-pas reprenant certaines procédures de façon imagée. L'utilisateur serait alors guidé vers les actions à réaliser.
- **Gain de performance et de poids.** Comme évoqué précédemment, la partie "app" est un site internet que l'on a intégré comme une application de bureau. Pour gagner en performance et en poids, il est possible d'héberger ce site internet sur un serveur Node.js qui se charge du calcul, et qui traite les informations depuis et vers la tablette.

Ces points ne sont bien entendu pas exhaustifs, mais cela représente des pistes d'amélioration intéressantes.

Cela conclut la section sur l'activité en entreprise. Dans la section suivante, nous traiterons des **bilans de la solution SPL**, de l'**expérience professionnelle**, et des **apports aussi bien pour l'entreprise que pour ma formation de Bachelor de Technologie**.

Conclusions

Annexes

Annexe 1 - Plan du site

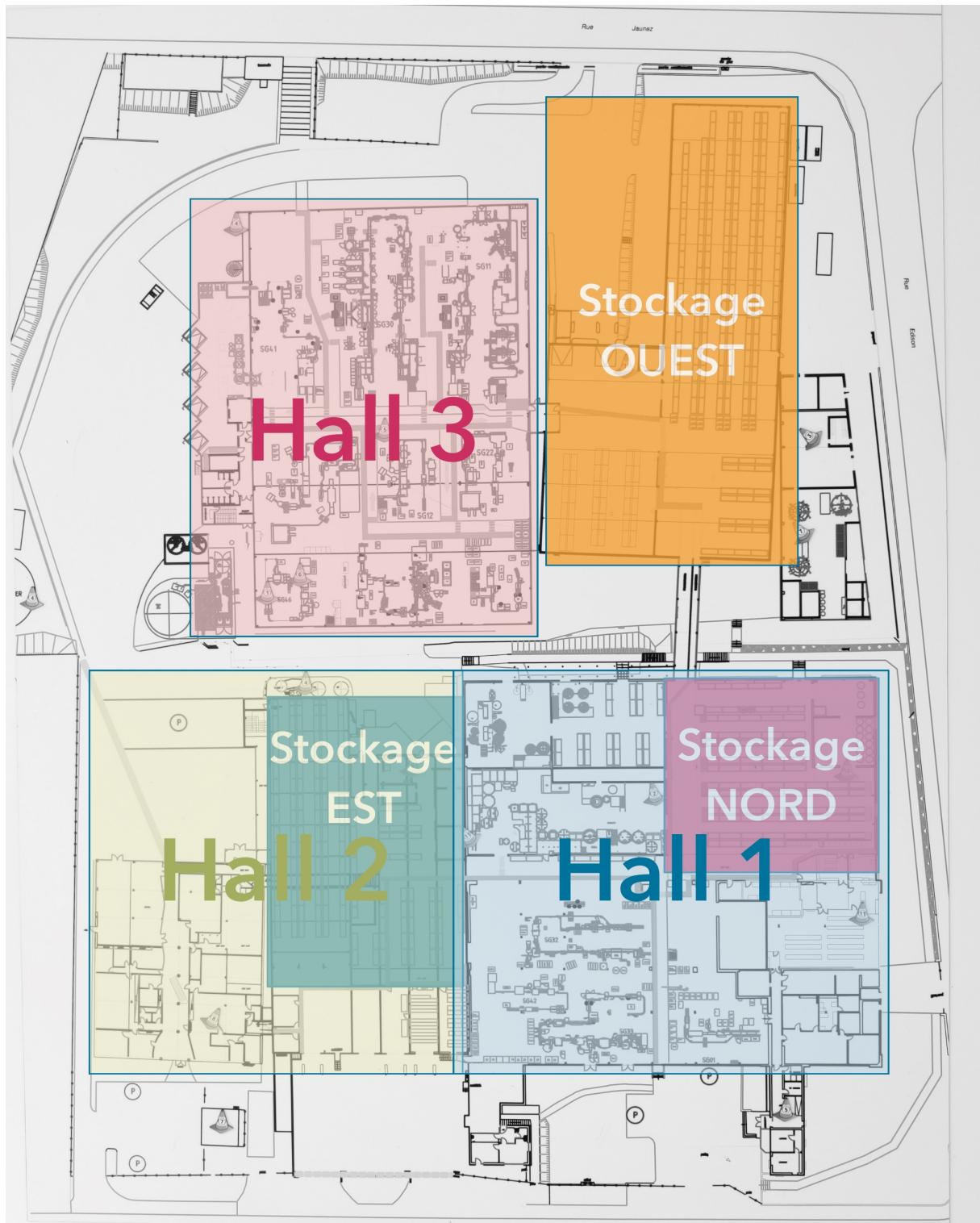


FIG. 14 : Plan du site

Annexe 2 - Organigramme de l'entreprise

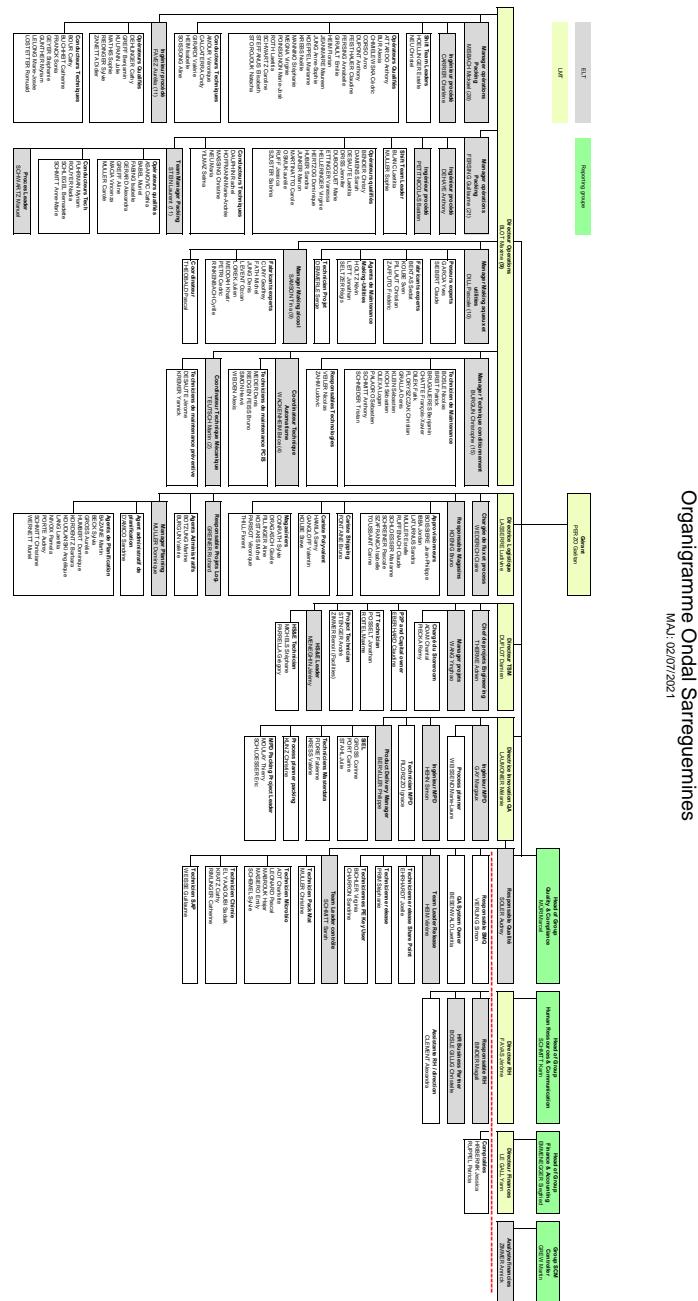


Fig. 15 : Organigramme de l'entreprise Ondal

Note : Seuls les employés en *Contrat à Durée Indéterminée (CDI)* sont représentés.